NASA-CR-191344

SBIR- 05.01-1167
release date 3/02/92

P.38

Radiation Monitoring Devices, Inc.
44 Hunt Street
Watertown, MA 02172

A PROTOTYPE NOVEL SENSOR FOR AUTONOMOUS, SPACE BASED ROBOTS - PHASE II

Final Report
July 1990

By:
M. R. Squillante, Ph.D.
L. P. de Rochemont, M. Eng.
L. Cirignano, M.S.
P. Lieberman, Ph.D
M. S. Soller, M.S.

Submitted to:
NASA Resident Office - JPL
4800 Oak Grove Drive
Pasadena, CA 91109
Contract No. NAS7-1032
Period: March 2, 1988 - July 2, 1990

(NASA-CR-191344) A PROTOTYPE NOVEL SENSOR FOR AUTONOMOUS, SPACE BASED ROBOTS, PHASE 2 Final Report No. 9, 2 Mar. 1988 - 2 Jul. 1990 (Radiation Monitoring Devices) 38 p

N93-19978

Unclas

G3/37 0143390

ORIGINAL PAGE IS OF POOR QUALITY

TABLE OF CONTENTS

A Prototype Novel Sensor for Autonomous, Space Based Robots

I. Introduction

The goal of this program was to develop new sensing capabilities for autonomous robots operating in space. Information gained by the robot using these new capabilities would be combined with other information gained through more traditional capabilities, such as video, to help the robot characterize its environment as well as to identify known or unknown objects that it encounters. Several sensing capabilities using nuclear radiation detectors and backscatter technology were investigated.

The result of this research has been the construction and delivery to NASA of a prototype system with three capabilities for use by autonomous robots. The primary capability was the use of beta particle backscatter measurements to determine the average atomic number (Z) of an object. This give the robot a powerful tool to differentiate objects which may look the same, such as objects made out of different plastics or other light weight materials.

In addition, the same nuclear sensor used in the backscatter measurement can be used as a nuclear spectrometer to identify sources of nuclear radiation that may be encountered by the robot, such as nuclear powered satellites. A complete nuclear analysis system is included in the software and hardware of the prototype system built in Phase II of this effort.

Finally, a method to estimate the radiation dose in the environment of the robot has been included as a third capability. Again the same nuclear sensor is used in a different operating mode and with different analysis software. Each of these capabilities are described below.

II. The Prototype System

The system consists of a custom made, highly sensitive CdTe nuclear sensor, a small isotopic source of beta particle radiation, a special collimator/detector package, a preamplifier, a personal computer, a multichannel analyzer, needed interfacing electronic circuitry, and all of the needed software.

A. Description of System Hardware

1. CdTe sensors

CdTe is a semiconductor material which can be used to fabricate very sensitive nuclear spectrometers which can operate at and above room temperature. CdTe was chosen because it high sensitivity allowed the entire sensing head to be small and because, by using CdTe no cryogenic coolant was necessary. In addition, these sensors have been used on previous space missions and have demonstrated there long term "space-worthyness".

2. The Beta Particle Source

During the program, a method was developed that allowed for the accurate determination of the average atomic number (Z) of a target independent of the density or chemical makeup of the source. This determination was done using beta particle backscatter. Beat particle scatter off electrons in a target. The number of electrons in the target is directly related to the atomic number of the constituent atoms and of the density. The amount of particles scattered is related to the density and the spectrum of the backscattered particles is related only to the average Z. Thus, by measuring the spectrum, the average Z can be determined. In principal, the density can be determined also by measuring the flux of backscattered particles. However, this is very difficult to calibrate and is a strong function of the distance from the source, the geometry of the measurement and to some extent the size and shape of the target. The backscatter spectrum is not affected by these factors.

Strontium-90 , ^{90}Sr, was chosen as the beta particle source because its long half life allows for stability and low maintenance and its high maximum energy, 2.7 MeV permits deeper interrogation into targets than would be possible with lower energy sources.

Figure 1 shows the results obtained. Clearly, plastics and elements are distinguishable by there atomic number. The relationships held for compounds also.

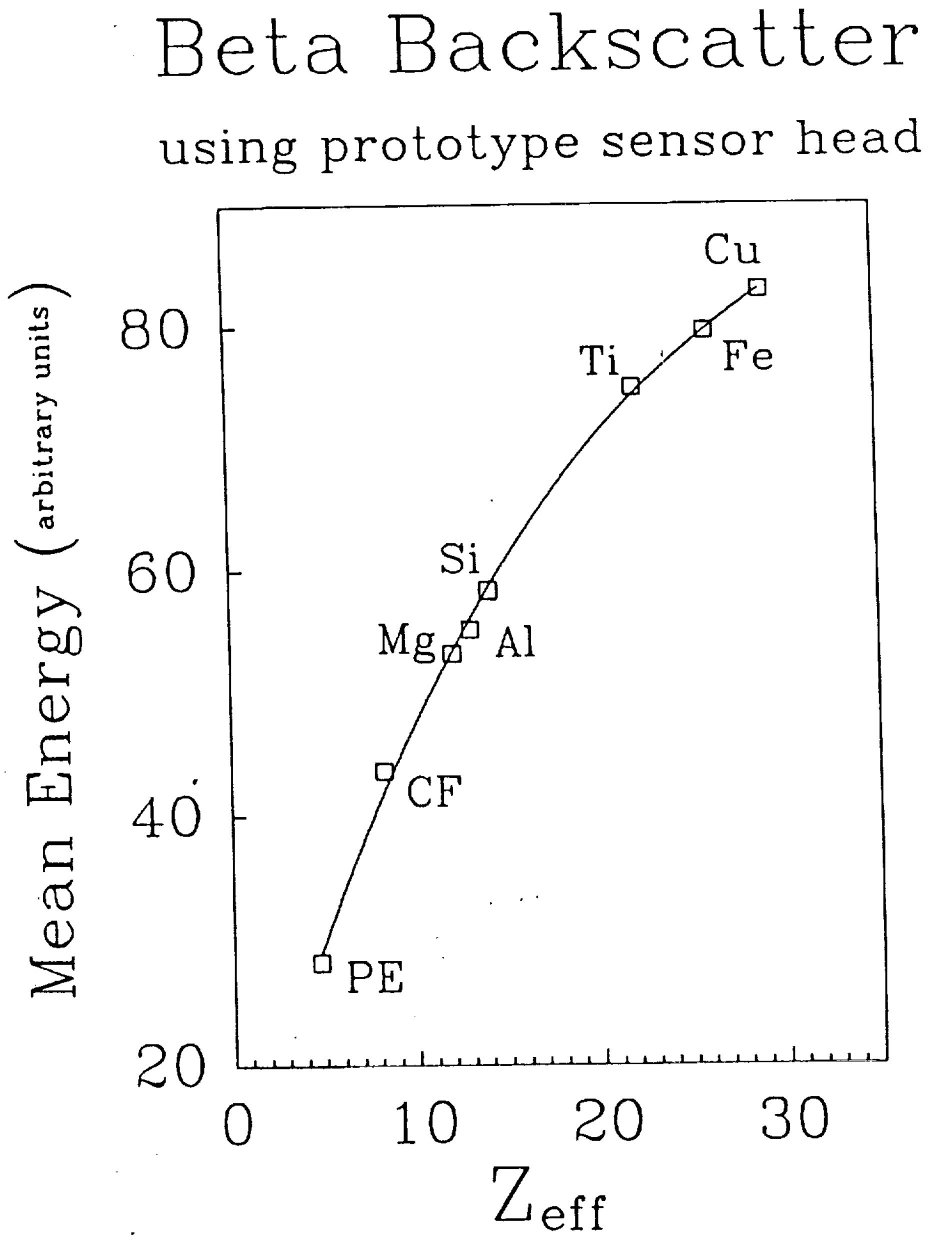


Figure 1. Experimental results on measuring Z_{eff}.

3. The sensor head

Figure 2 shows a schematic of the sensor head that was developed for the prototype. **Figure 3** and **4** show photographs of the unit. The primary design considerations were that it must 1) be compact and light weight, 2) be safe to operators in the vicinity, and 3) permit accurate measurement of the target of interest. The design satisfies all of these requirements.

The head consists of a Delrin plastic sensor head, a CdTe detector, a ^{90}Sr isotopic source contained in a graphite holder, and a lead shield located directly between the source and detector. This head attaches directly to a compact PCT-3 preamplifier manufactured by RMD, Inc. The preamplifier plugs into the interface ports on the personal computer.

D. Description The Computer System and System Operation

The prototype makes use of a standard, IBM-type, portable computer. A commercial multichannel analyzer, "The Norland 5000" board by Norland Corp., has been added. All of the user interface software has been written in BASIC. Three capabilities are accesses from the keyboard through the menu screen that appears on boot up.

1. Measurement of Z Effective

This primary capability provide the robot with the average atomic number of the target. With the isotopic source provided, the system will measure the spectrum backscattered from the target and determine Z_{eff}. Z_{eff} is close to, not exactly the simple arithmetic average of the constituent elements but related to the atomic numbers of the them by the following equation:

$$Z_{eff} = \sum (W_i / A_i) Z_i^2 / \sum (W_i / A_i) Z_i$$

where W_i is the weight fraction of the i^{th} element, A_i is its atomic weight and Z_i is its atomic number.

This function of the prototype is selected from the keyboard by first choosing the Zeff function from menu on the screen, as described in the attached operators manual.

2. Dose Rate Measurement

The gamma ray radiation dose rate dose at the location of the robot is

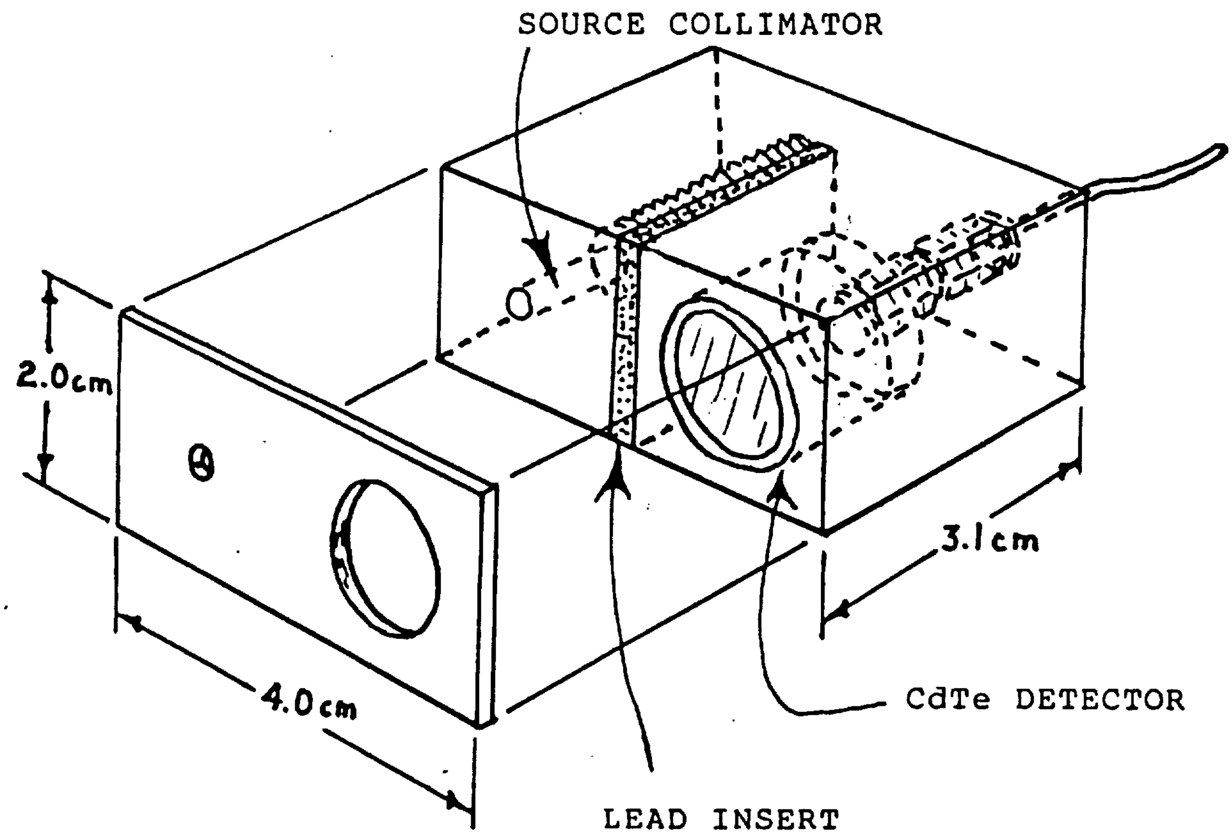


Figure 2. Sketch of the backscatter sensor head.

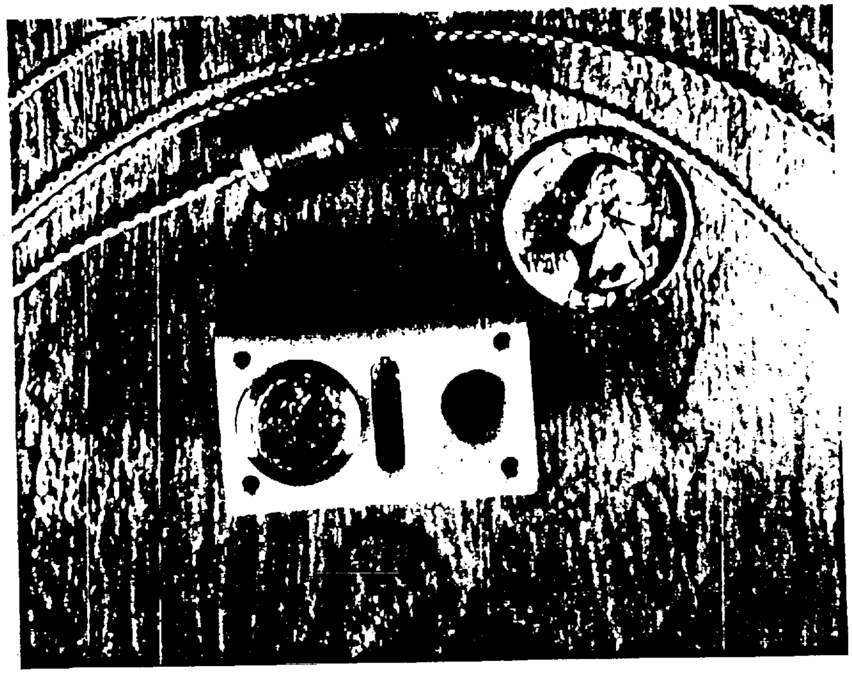


Fig.3 . Photograph of the prototype backscatter sensor with the front faceplate missing. Circular object is the front of the beta sensitive CdTe detector, oblong object is the lead x-ray shield, smaller hole is for the beta source.

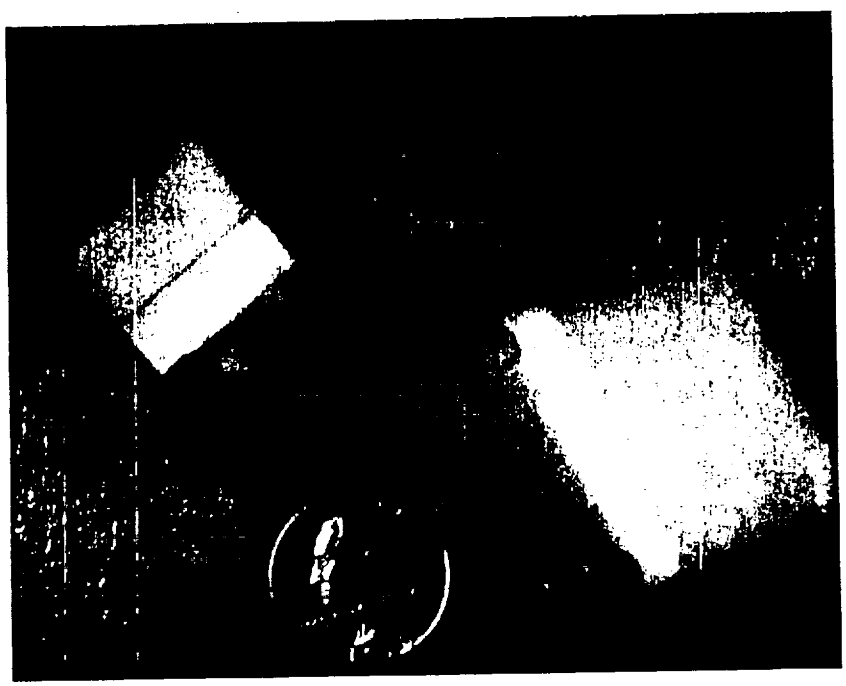


Fig.4 . Photograph of the adjustable and fixed sensor heads with the faceplate collimators in place.

determined using an empirical method based on the response on the CdTe detector to gamma rays. The detector measures the average energy of many rays striking it, the average energy deposited in the detector is measured and the "tissue equivalent" dose is estimated using a predetermined calibration curve. Experiments performed with this detector and software showed the dose rate could be determined with about a 20% accuracy over the energy range of 100 keV to about 3 Mev.

Measuring the dose rate is achieved by selecting "Personal Dosimeter" at the menu screen.

3. Nuclear Spectrum Analyzer.

The third capability included in the prototype is nuclear pulse height analysis. Incorporated into the prototype is a complete Norland 5000 multichannel analyzer. This feature is accessed by choosing "Norland" from the menu. The details for operating this analyzer are described in detail in the operators manual which accompanies the prototype.

III. Summary

This SBIR program has been successfully completed and has met the technical goals of developing new sensing capabilities for autonomous space-based robots. A complete and fully functional prototype has been assembled, tested and delivered to NASA at JPL in Pasadena. This prototype demonstrate several possible capabilities which should provide useful senses to robot designers for future generations of robots.

BETA BACKSCATTER INSTRUMENT FOR ATOMIC NUMBER DETERMINATION

INSTRUCTION MANUAL

JULY, 1990

RADIATION MONITORING DEVICES, INC.
44 HUNT STREET,
WATERTOWN, MA 02172

I. INTRODUCTION

The beta backscatter instrument is designed to provide an accurate measurement of effective atomic number (Z_{eff}) of a target in under one minute at a distance of 4 cm. from the sensor head. Z_{eff} is defined as:

$$Z_{eff} = \Sigma(w_i/A_i)Z_i^2/\Sigma(w_i/A_i)Z_i$$

where w_i = weight fraction of the i^{th} element, A_i = the atomic weight of the i^{th} element and Z_i = the atomic number of the i^{th} element.

The portable lab prototype consists of: (1) a Delrin plastic sensor head, (2) CdTe beta detector, (3) $^{90}Sr/^{90}Y$ beta source in a threaded graphite source holder, (4) standard PCT-3 preamplifier, (5) portable personal computer which includes specially designed expansion boards for providing detector bias voltage and for storing and analyzing the data, (6) cables. The entire system is compact, powered by 110VAC and very convenient to operate in the laboratory environment.

II. INITIAL SET-UP

Since the computer has been provided with the expansion boards already installed, the only interconnection to be made is to connect the preamplifier cord to the 9-pin connector at the rear of the amplifier/bias expansion board. In addition, it is worth checking to see that the short BNC cable which runs between "AMP OUT" on the first board and "MCA IN" on the second board is still firmly connected after shipping.

Although there are internal settings on the amplifier expansion board which can be used to adjust the gain and detector bias, these settings have been optimized at RMD and, in general, should not be altered.

III. OPERATION

To run the Z_{eff} program, type "ZEFF". The main menu allows the user to select from the five options listed below:

[ ] INSTRUMENT SETTINGS
[ ] BACKGROUND ROUTINES
[ ] CALIBRATION ROUTINES
[ ] MEASUREMENT ROUTINE
[ ] EXIT THE PROGRAM

A. Instrumernt Settings

This option allows one to set the spectrum Region Of Interest (ROI), background, data collection, and calibration acquisition periods. Normally the ROI settings should not be changed from the default settings.

Once the INSTRUMENT SETTINGS have been chosen the operator should return to the main MENU.

B. Background Routines

Before using either the calibration or measurement routines a background spectrum must be stored. Therefore the BACKGROUND ROUTINES should be chosen from the main MENU.

In order to acquire background data, the source holder should first be installed in the sensor head. The sensor head should then be positioned such that there are no objects within two meters of the front of the sensor head to backscatter electrons toward the detector. Once background data has been acquired the spectrum can be saved to disk from the "BACKGROUND ROUTINES" menu. Once saved, the background spectrum can be recalled for use at a later time.

Note that the background spectrum may depend on the position of the source in the sensor head. Therefore the background spectrum corresponding to the particular source position should be used.

C. Calibration Routines

Values for calibration constants, A, B, and C, are required in order to determine Z_{eff} of a given target. Therefore one of the options from the calibration mode must be exercised before proceeding to the MEASUREMENT ROUTINES.

Choosing the option "USE DEFAULT CALIBRATION" automatically sets A, B, and C to values which were preset at RMD. Depending on the particular experimental arrangement, these values may or may not yield accurate results. In order to ensure accurate results for a new experimental set-up, the "RECALIBRATE " option should be chosen. A minimum of three targets with known atomic numbers are required for calibration. For each calibration target the effective atomic number should be entered and then a spectrum taken. After the backscatter spectrum for the last calibration target has been measured, calculated values for A, B, and C are displayed. Effective atomic number values, calculated using the determined values for A, B, and C, are also displayed in order to evaluate the accuracy of the calibration. When the calibration is complete the <Esc> key returns the system to the main menu. The system is now ready to determine effective atomic number of unknown targets.

The calibration constants for a particular configuration can be used later without going through the recalibration procedure. The "ENTER NEW CALIBRATION CONSTANTS" option should be chosen. Using this option allows one to directly enter the values for A, B, and C.

Note that for best results, one should choose calibration samples which cover the range of atomic numbers of interest. For example, if samples to be measured have Z_{eff} values between 6 and 30, then calibration samples with Z_{eff} values of 4.7 (Polyethylene), 13 (Aluminum), and 29 (Copper) would be appropriate.

D. Measurement Routine

The "MEASUREMENT ROUTINE" option should be chosen after appropriate background spectra and calibration data have been taken. Upon completion of acquiring data and calculating Z_{eff}, the software displays TOTAL COUNTS IN ROI, AVERAGE CHANNEL NUMBER, and EFFECTIVE ATOMIC NUMBER. When the next measurement is performed, these numbers are cleared and then updated at the completion of the calculation.

E. Exit The Program

While in the main menu, one can choose to quit the program and return to DOS. When in DOS, the computer is available to be used as a standard personal computer.

```
' *** HEADER *****************************************************************

' *************************************************************************
' *                                                                       *
' *                              ZEFF.BAS                                 *
' *                                                                       *
' *                            ( RMD Inc. )                               *
' *                                                                       *
' *                                                                       *
' *************************************************************************
' *                                                                       *
' *************************************************************************

' *** Arrays and Variables ***************************************************


DECLARE FUNCTION chanread& (A%, B%, C%)  ' READ DATA A CHANNEL AT A TIME
DECLARE SUB erasepha (A%, B%, C%, D%)    ' ERASES PHA MEMORY
DECLARE SUB setlld (lld%)                ' SETS LOWER LEVEL DISCRIMINATOR
DECLARE SUB setuld (uld%)                ' SETS UPPER LEVEL DISCRIMINATOR
DECLARE SUB setoffset (offset%, gain%)   ' SETS DIGITAL OFFSET AND CONVERSION GAI
DECLARE SUB acquire (A%, B%, C%)         ' CONTROLS ACQUISITION TIME

COMMON SHARED TIME%                      ' ACQUISITION TIME

DIM backgroundarray&(1024)          ' ARRAY WHERE BACKGROUND IS STORED
DIM dataarray&(1024)                ' ARRAY WHERE DATA IS STORED
DIM processedarray!(1024)           ' ARRAY FOR PROCESSED DATA
DIM Z(20)                           ' CALIBRATION Zeff ARRAY
DIM E(20)                           ' CALIBRATION <CH#> ARRAY

DIM instdata(20)                    ' INSTRUMENT SETTINGS

DATA = 0                            ' FLAG, 0 = NO DATA ACQUIRED
BKGND = 0                           ' FLAG, 0 = NO BACKGROUND INFORMATION
CAL = 0                             ' FLAG, 0 = NO CALIBRATION INFORMATION
KEY(12) ON


instdata(10) = 15                   ' ROI LOW
instdata(12) = 800                  ' ROI HIGH
instdata(14) = 100                  ' BKGND MEASUREMENT PERIOD (TYME)
instdata(16) = 15                   ' DATA MEASUREMENT PERIOD (TIME)
instdata(18) = 30                   ' CALIBRATION MEASUREMENT PERIOD

A = -.2060272                       ' DEFAULT VALUE FOR QUADRATIC COEFF. A
B = 14.07163                        '                 "                  B
C = 27.1889                         '                 "                  C

' ***************************************************************************

BOX: ' *********************************************************************
        ' *
        ' *** CONSTRUCTS ON SCREEN DISPLAY BORDERS ****************************
        ' *
```

```
        ' ************************************************************************
CLS :
FOR V = 1 TO 23
        IF V = 1 OR V = 6 OR V = 21 OR V = 23 THEN
                FOR H = 1 TO 80
                        LOCATE V, H: PRINT "ë"          ' HORIZONTAL LINES
                NEXT H
        END IF
                FOR H = 1 TO 80 STEP 79
                        LOCATE V, H: PRINT "¤"          ' VERTICAL LINES
                NEXT H
        IF V = 1 THEN LOCATE 1, 1: PRINT "è": LOCATE 1, 80: PRINT "£"
        IF V = 6 THEN LOCATE 6, 1: PRINT "ä": LOCATE 6, 80: PRINT "¿"
        IF V = 21 THEN LOCATE 21, 1: PRINT "ä": LOCATE 21, 80: PRINT "¿"
        IF V = 23 THEN LOCATE 23, 1: PRINT "à": LOCATE 23, 80: PRINT "¥"
NEXT V
LOCATE 2, 38: PRINT "ZEFF"
LOCATE 3, 20: PRINT "TO MEASURE EFFECTIVE Z BY BETA BACKSCATTER"
LOCATE 5, 3: PRINT ""; DATE$; ""
LOCATE 5, 71, 0: PRINT ""; TIME$; ""

' ****************************************************************************

ACKNOWLEDGEMENT: ' *************************************************************
                          ' *
                          ' *** PRINTS ON-SCREEN ACKNOWLEDGEMENT ***************
                          ' *
                          ' ***************************************************

LOCATE 8, 23, 0: PRINT "WRITTEN BY KENNETH J. BOROWSKI, Ph.D."
LOCATE 12, 24: PRINT "RADIATION MONITORING DEVICES, INC."
LOCATE 14, 34: PRINT "44 HUNT STREET"
LOCATE 16, 26: PRINT "WATERTOWN, MASSACHUSETTS 02172"
LOCATE 18, 35: PRINT "617-926-1167"
LOCATE 20, 33: PRINT "COPYRIGHT 1989"
LOCATE 22, 29: PRINT "HIT ANY KEY TO CONTINUE"
DO
        LOCATE 5, 71, 0: PRINT ""; TIME$; ""        ' update real time
LOOP WHILE INKEY$ = ""

' ****************************************************************************

MENU: ' ***********************************************************************
        ' *
        ' *** MAIN MENU ******************************************************
        ' *
        ' ********************************************************************

GOSUB FLUSH

LOCATE 10, 11: PRINT " [ ]      INSTRUMENT SETTINGS"
LOCATE 12, 11: PRINT " [ ]      BACKGROUND ROUTINES"
LOCATE 14, 11: PRINT " [ ]      CALIBRATION ROUTINE"
LOCATE 16, 11: PRINT " [ ]      MEASUREMENT ROUTINE"
LOCATE 18, 11: PRINT " [ ]      EXIT THE PROGRAM"
```

```
LOCATE 22, 14: PRINT "   USE <UP> OR <DOWN> ARROWS THEN <ENTER> TO SELECT."

        D1 = 18

MLIM: ' *** MANAGES THE LIMITS OF TRAVEL OF THE CURSOR *************************
        IF D1 > 18 THEN D1 = 10 ELSE IF D1 < 10 THEN D1 = 18
        LOCATE D1, 13, 1, 0, 13
        D1 = CSRLIN
        A$ = INKEY$: IF A$ = CHR$(13) THEN GOTO MCURSVAL ELSE IF A$ = "8" THEN D
        IF A$ = "2" THEN D1 = D1 + 2: GOTO MLIM
        IF LEN(A$) = 2 THEN A$ = RIGHT$(A$, 1) ELSE GOTO MLIM
        IF A$ = "H" THEN D1 = D1 - 2: GOTO MLIM
        IF A$ = "P" THEN D1 = D1 + 2: GOTO MLIM
        GOTO MLIM
        D1 = D1 + 2: GOTO MLIM

MCURSVAL: ' *** FINDS THE VALUE OF THE LINE AT THE CURSOR LOCATION *************
        D1 = CSRLIN
        IF CSRLIN = 10 THEN GOSUB INSTRUMENT          ' INSTRUMENT SETTINGS
        IF CSRLIN = 12 THEN GOSUB BACKGROUND          ' BACKGROUND ROUTINES
        IF CSRLIN = 14 THEN GOSUB CALIB               ' CALIBRATION ROUTINE
        IF CSRLIN = 16 THEN GOSUB MEASURE             ' MEASUREMENT MODE
        IF CSRLIN = 18 THEN BEEP: CLS : END           ' EXIT FROM THE PROGRAM
        GOTO MENU

' *************************************************************************

RETURN  ' *****************************************************************

INSTRUMENT: ' *************************************************************
                    ' *
                    ' *** INSTRUMENT SETTINGS MENU ********************************
                    ' *
                    ' *************************************************************
GOSUB FLUSH

LOCATE 7, 22: PRINT "*** INSTRUMENT SETTINGS ROUTINE ***"
LOCATE 10, 3: PRINT " [ ]  SET ROI LOWEST CHANNEL"; TAB(50); instdata(10)
LOCATE 12, 3: PRINT " [ ]  SET ROI HIGHEST CHANNEL"; TAB(50); instdata(12)
LOCATE 14, 3: PRINT " [ ]  SET BACKGROUND ACQUISITION PERIOD"; TAB(50); instdata
LOCATE 16, 3: PRINT " [ ]  SET DATA ACQUISITION PERIOD"; TAB(50); instdata(16);
LOCATE 18, 3: PRINT " [ ]  SET CALIBRATION ACQUISITION PERIOD"; TAB(50); instdat
LOCATE 20, 3: PRINT " [ ]  RETURN TO MENU "
LOCATE 22, 14: PRINT "   USE <UP> OR <DOWN> ARROWS THEN <ENTER> TO SELECT."
        D1 = 20                 ' SETS CURSOR TO PROPER

ILIM: ' *** MANAGES THE LIMITS OF TRAVEL OF THE CURSOR *************************
        IF D1 > 20 THEN D1 = 10 ELSE IF D1 < 10 THEN D1 = 20
        LOCATE D1, 5, 1, 0, 13
        D1 = CSRLIN
        A$ = INKEY$: IF A$ = CHR$(13) THEN GOTO ICURSVAL ELSE IF A$ = "8" THEN D
        IF A$ = "2" THEN D1 = D1 + 2: GOTO ILIM
        IF LEN(A$) = 2 THEN A$ = RIGHT$(A$, 1) ELSE GOTO ILIM
        IF A$ = "H" THEN D1 = D1 - 2: GOTO ILIM
        IF A$ = "P" THEN D1 = D1 + 2: GOTO ILIM
```

```
        GOTO ILIM
        D1 = D1 + 2: GOTO ILIM

ICURSVAL: ' *** FINDS THE VALUE OF THE LINE AT THE CURSOR LOCATION *************
        D1 = CSRLIN
        IF CSRLIN = 10 THEN GOSUB ICHANGE           ' ROI LOW
        IF CSRLIN = 12 THEN GOSUB ICHANGE           ' ROI HIGH
        IF CSRLIN = 14 THEN GOSUB ICHANGE           ' BKGND PERIOD
        IF CSRLIN = 16 THEN GOSUB ICHANGE           ' DATA PERIOD
        IF CSRLIN = 18 THEN GOSUB ICHANGE           ' CALIB. PERIOD
        IF CSRLIN = 20 THEN GOTO MENU               ' RETURN MAIN MENU
        GOTO INSTRUMENT

ICHANGE:
FOR I = 50 TO 70
        LOCATE D1, I, 0: PRINT "  "
NEXT I
        IF D1 = 10 THEN LOCATE D1, 60, 0: PRINT "(1 TO 1024)"
        IF D1 = 12 THEN LOCATE D1, 60, 0: PRINT "(1 TO 1024)"
        IF D1 = 14 THEN LOCATE D1, 58, 0: PRINT "(1 TO 10,000 SECONDS)"
        IF D1 = 16 THEN LOCATE D1, 58, 0: PRINT "(1 TO 10,000 SECONDS)"
        IF D1 = 18 THEN LOCATE D1, 58, 0: PRINT "(1 TO 10,000 SECONDS)"

LOCATE D1, 49: INPUT SETTINGS

IF D1 = 10 THEN
        IF SETTINGS < 1 OR SETTINGS > 1024 THEN BEEP: GOTO ICHANGE
END IF
IF D1 = 12 THEN
        IF SETTINGS < 1 OR SETTINGS > 1024 THEN BEEP: GOTO ICHANGE
END IF
IF D1 = 14 THEN
        IF SETTINGS < 1 OR SETTINGS > 10000 THEN BEEP: GOTO ICHANGE
END IF
IF D1 = 16 THEN
        IF SETTINGS < 1 OR SETTINGS > 10000 THEN BEEP: GOTO ICHANGE
END IF
IF D1 = 18 THEN
        IF SETTINGS < 1 OR SETTINGS > 10000 THEN BEEP: GOTO ICHANGE
END IF

instdata(D1) = SETTINGS          ' RESETS INSTRUMENT

RETURN

' ***********************************************************************

BACKGROUND: ' ***********************************************************
                  ' *
                  ' *** BACKGROUND ROUTINES *************************************
                  ' *
                  ' *************************************************************

BKGNDFILE$ = "BKGND.DAT"
```

```
GOSUB FLUSH

LOCATE 8, 26: PRINT "**** BACKGROUND ROUTINES ****"
LOCATE 11, 11: PRINT " [ ]      RECALL BACKGROUND DATA FROM DISK"
LOCATE 13, 11: PRINT " [ ]      SAVE BACKGROUND DATA ON DISK"
LOCATE 15, 11: PRINT " [ ]      ACQUIRE BACKGROUND DATA"
LOCATE 17, 11: PRINT " [ ]      RETURN TO MAIN MENU"
LOCATE 22, 14: PRINT "   USE <UP> OR <DOWN> ARROWS THEN <ENTER> TO SELECT."
        D1 = 17

BAKDLIM: ' *** MANAGES THE LIMITS OF TRAVEL OF THE CURSOR ***

IF D1 > 17 THEN D1 = 11 ELSE IF D1 < 11 THEN D1 = 17
LOCATE D1, 13, 1, 0, 13
D1 = CSRLIN
A$ = INKEY$: IF A$ = CHR$(13) THEN GOTO BAKCURSVAL ELSE IF A$ = "8" THEN D1 = D1
IF A$ = "2" THEN D1 = D1 + 2: GOTO BAKDLIM
IF LEN(A$) = 2 THEN A$ = RIGHT$(A$, 1) ELSE GOTO BAKDLIM
IF A$ = "H" THEN D1 = D1 - 2: GOTO BAKDLIM
IF A$ = "P" THEN D1 = D1 + 2: GOTO BAKDLIM
GOTO BAKDLIM
D1 = D1 + 2: GOTO BAKDLIM

BAKCURSVAL: ' *** FINDS THE VALUE OF THE LINE AT THE CURSOR LOCATION ***

D1 = CSRLIN
IF CSRLIN = 11 THEN GOSUB BKGNDCALL          ' RECALL BACKGROUND DATA
IF CSRLIN = 13 THEN GOSUB BKGNDSAVE          ' SAVE BACKGROUND DATA
IF CSRLIN = 15 THEN GOSUB BKGNDMEASURE       ' MEASURE BACKGROUND
IF CSRLIN = 17 THEN RETURN

GOTO BACKGROUND

BKGNDCALL: ' ****************************************************************
                    ' *
                    ' *** RECALL BACKGROUND DATA FROM DISK  **********************
                    ' *
                    ' ***************************************************************
GOSUB FLUSH

LOCATE 8, 26: PRINT "**** BACKGROUND DATA RECALL ****"
BKGNDFILE$ = "BKGND.DAT"

BKGNDCHK: ' *** CHECK FOR BACKGROUND DATA ***

IF PNT = 1 THEN
        BEEP: LOCATE 14, 26: PRINT "BACKGROUND DATA ALREADY IN PLACE"
        LOCATE 22, 31: INPUT "REPLACE DATA (Y/N) "; C$
        IF C$ = "Y" OR C$ = "y" THEN GOTO SELBKGNDFIL
        IF C$ = "N" OR C$ = "n" THEN GOTO BACKGROUND
        GOTO BKGNDCHK
END IF
BEEP: LOCATE 22, 14: INPUT "DO YOU WANT TO RETRIEVE A BACKGROUND DATA FILE (Y/N)
IF r$ = "Y" OR r$ = "y" THEN
```

```
        GOTO SELBKGNDFIL
ELSEIF r$ = "N" OR r$ = "n" THEN
        GOTO BACKGROUND
END IF
GOTO BKGNDCHK

SELBKGNDFIL: ' *** SELECT BACKGROUND DATA FILE ON DISK ***

GOSUB FLUSH

LOCATE 8, 26: PRINT "**** BACKGROUND DATA RECALL ****"
LOCATE 12, 19: PRINT "BACKGROUND DATA FILE [ DEFAULT = "; BKGNDFILE$; " ] ";
LOCATE 22, 16: PRINT "<ENTER> FOR DEFAULT FILE NAME OR TYPE IN OTHER NAME"
LOCATE 14, 48: INPUT ; A$
IF A$ <> "" THEN BKGNDFILE$ = A$
IF BKGNDFILE$ = "" THEN BEEP: GOTO SELBKGNDFIL
FOR J = 25 TO 75
        LOCATE 14, J, 0: PRINT "  "
NEXT J                                                  ' CLEAR LINE 14
FOR J = 15 TO 75
        LOCATE 12, J, 0: PRINT "  "
NEXT J                                                  ' CLEAR LINE 12
LOCATE 12, 19: PRINT "BACKGROUND DATA FILE [ DEFAULT = "; BKGNDFILE$; " ] ";
CLOSE #1
OPEN BKGNDFILE$ FOR APPEND AS #1
XIMPUTFILE = LOF(1)
CLOSE #1
IF XIMPUTFILE <= 2 THEN
        KILL BKGNDFILE$
        BEEP: LOCATE 14, 29: PRINT BKGNDFILE$; " NOT FOUND ON DISK"
        FOR J = 2 TO 78
                LOCATE 22, J, 0: PRINT " "
        NEXT J                                          ' CLEARS COMMAND LINE
        LOCATE 22, 29: PRINT "HIT ANY KEY TO CONTINUE"
        DO
        LOCATE 5, 71: PRINT ""; TIME$; ""
        LOOP WHILE INKEY$ = ""                          ' UPDATE REAL TIME
        GOTO BKGNDCALL
END IF

LOABKGNDFIL: ' *** LOADS BACKGROUND DATA FILE ***

OPEN BKGNDFILE$ FOR INPUT AS #1
INPUT #1, BKGNDDATE$, BKGNDTIME$, BKGNDTIME
FOR I = 1 TO 6
        INST = (6 + (2 * I))
        INPUT #1, instdata(INST)
NEXT I
FOR I = 1 TO 1024
        INPUT #1, backgroundarray&(I)
NEXT I

instdata(14) = BKGNDTIME

CLOSE #1
```

```
BKGND = 1          ' SETS BACKGROUND PRESENT FLAGS

RETURN  ' ********************************************************************

BKGNDSAVE: ' *****************************************************************
                 ' *                                                          *
                 ' *** SAVE BACKGROUND DATA TO DISK ****************************
                 ' *                                                          *
                 ' ************************************************************
GOSUB FLUSH

IF BKGND = 0 THEN
        BEEP: LOCATE 10, 22: PRINT "*** NO BACKGROUND COLLECTED OR INSTALLED ***
        LOCATE 19, 24: PRINT "HIT ANY KEY TO RETURN TO BACKGROUND MENU"
        DO
        LOCATE 5, 71: PRINT ""; TIME$; ""          ' UPDATE REAL TIME
        LOOP WHILE INKEY$ = ""
        GOTO BACKGROUND

END IF

DSSK: ' *** DISK OUTPUT ROUTINE ***

BEEP

GOSUB FLUSH

BKGNDFILID: ' *** FILe IDentification ***

LOCATE 8, 26: PRINT "**** BACKGROUND DATA SAVER ****"
LOCATE 22, 15: INPUT "DO YOU WANT TO SAVE A BACKGROUND DATA FILE (Y/N) "; D$
IF D$ = "N" OR D$ = "n" THEN GOTO BACKGROUND
IF D$ = "Y" OR D$ = "y" THEN GOTO OWTDEFAULT
GOTO DSSK

OWTDEFAULT: ' *** OUTPUT FILE DEFAULT NAME ***

LOCATE 12, 19: PRINT "BACKGROUND DATA FILE [ DEFAULT = "; BKGNDFILE$; " ]";
FOR J = 42 TO 75
        LOCATE 14, J, 0: PRINT " "
NEXT J
LOCATE 22, 15: PRINT "<ENTER> FOR DEFAULT FILE NAME OR TYPE IN OTHER NAME  "
LOCATE 14, 46: INPUT ; A$
IF A$ <> "" THEN BKGNDFILE$ = A$
IF BKGNDFILE$ = "" THEN BEEP: GOTO BKGNDFILID
FOR J = 25 TO 75
        LOCATE 14, J, 0: PRINT " "
NEXT J
LOCATE 12, 19: PRINT "BACKGROUND DATA FILE [ DEFAULT = "; BKGNDFILE$; " ]";
CLOSE #1
OPEN BKGNDFILE$ FOR APPEND AS #1
XOUTPUTFILE = LOF(1)
CLOSE #1
IF XOUTPUTFILE > 2 THEN BEEP: LOCATE 14, 29: PRINT BKGNDFILE$; " ALREADY ON DISK
```

```
OVWRT: ' *** OVERWRITE ***

  IF XOUTPUTFILE <= 2 THEN GOTO SAVBKGNDFIL
          FOR J = 7 TO 75
                  LOCATE 22, J, 0: PRINT " "
          NEXT J                                                    ' CLEAR COMMAND LINE
          LOCATE 22, 32, 0: INPUT "OVER WRITE (Y/N) "; B$
          IF B$ = "Y" OR B$ = "y" THEN
          KILL BKGNDFILE$: GOTO SAVBKGNDFIL
 END IF
 IF B$ = "N" OR B$ = "n" THEN
          FOR J = 25 TO 75
                  LOCATE 14, J, 0: PRINT " "
          NEXT J
          GOTO DSSK                                                 ' CLEAR LINE 14
 END IF
          BEEP: GOTO OVWRT

SAVBKGNDFIL: ' *** SAVEFILE ***

BKGNDDATE$ = DATE$: BKGNDIME$ = TIME$

BKGNDTIME = instdata(14)
OPEN BKGNDFILE$ FOR OUTPUT AS #1
WRITE #1, BKGNDDATE$, BKGNDTIME$, BKGNDTIME
FOR I = 1 TO 6
        INST = (6 + (2 * I))
        WRITE #1, instdata(INST)
NEXT I
FOR I = 1 TO 1024
        WRITE #1, backgroundarray&(I)
NEXT I

CLOSE #1

RETURN ' ******************************************************************

BKGNDMEASURE: ' *************************************************************
                  ' *
                  ' *** CONTROLS BACKGROUND ACQUISITION ACTIVITIES ***********
                  ' *
GOSUB FLUSH       ' *****************************************************************

BKSCREEN: ' *** BACKGROUND SCREEN DISPLAY ***

LOCATE 7, 23: PRINT "*** BACKGROUND ACQUISITION MODE ***"
LOCATE 11, 5: PRINT "BACKGROUND PERIOD IS "; instdata(14); "SECONDS"
LOCATE 11, 42: PRINT "YOU MUST MAKE A SINGLE MEASUREMENT"
LOCATE 22, 3, 0: PRINT "POSITION PROBE, THEN HIT <SPACE BAR> TO BEGIN MEASUREMEN
DO
      LOCATE 5, 71, 0: PRINT ""; TIME$; ""              ' UPDATE REAL TIME
```

```
A$ = INKEY$
        IF A$ = CHR$(27) THEN GOTO MENU        ' INTERRUPT
        IF A$ = CHR$(32) THEN GOTO BKOK
LOOP

BKOK:

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "              ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 18, 0: PRINT "MEASUREMENT IN PROGRESS, WAIT FOR COMPLETION"

TIME% = instdata(14)

GOSUB NORLAND

' COME BACK FROM NORLAND

' MOVE DATA INTO BACKGROUNDARRAY

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "          ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 17, 0: PRINT "DATA TRANSFER IN PROGRESS, WAIT FOR COMPLETION"

FOR x% = 1 TO 1024
        backgroundarray&(x%) = chanread&(x%, &HF300, &HD000)
NEXT x%

BEEP

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "          ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 16: PRINT "MEASUREMENT OVER, HIT ANY KEY TO RETURN TO MENU"

DO
        LOCATE 5, 71, 0: PRINT ""; TIME$; ""   ' UPDATE REAL TIME
LOOP WHILE INKEY$ = ""

BKGND = 1                              ' BACKGROUND DATA FLAG

GOTO MENU

RETURN ' ***********************************************************************

MEASURE: ' *********************************************************************
          ' *
          ' *** CONTROLS DATA ACQUISITION ACTIVITIES *************************
          ' *
          ' *******************************************************************
```

```
GOSUB FLUSH
 IF BKGND = 0 THEN
    LOCATE 11, 17: PRINT " MUST HAVE BACKGROUND SPECTRUM BEFORE TAKING DATA."
    LOCATE 22, 18, 0: PRINT " HIT ANY KEY TO RETURN TO MAIN MENU."
    DO
         LOCATE 5, 71, 0: PRINT ""; TIME$; ""
    LOOP WHILE INKEY$ = ""
    GOTO MENU
 END IF
 IF CAL = 0 THEN
    LOCATE 11, 17: PRINT " MUST CALIBRATE SYSTEM BEFORE TAKING DATA."
    LOCATE 22, 18, 0: PRINT " HIT ANY KEY TO RETURN TO MAIN MENU."
    DO
         LOCATE 5, 71, 0: PRINT ""; TIME$; ""
    LOOP WHILE INKEY$ = ""
    GOTO MENU
 END IF

MEASURESCREEN: ' *** MEASUREMENT SCREEN DISPLAY ***

LOCATE 8, 26: PRINT "*** DATA ACQUISITION MODE ***"
LOCATE 11, 22: PRINT "MEASUREMENT PERIOD IS "; instdata(16); "SECONDS"
LOCATE 22, 3, 0: PRINT "POSITION PROBE, THEN HIT <SPACE BAR> TO BEGIN MEASUREMEN
DO
        LOCATE 5, 71, 0: PRINT ""; TIME$; ""          ' UPDATE REAL TIME
A$ = INKEY$
        IF A$ = CHR$(27) THEN GOTO MENU          ' INTERRUPT
        IF A$ = CHR$(32) THEN GOTO MSOK
LOOP

MSOK:
LOCATE 16, 52: PRINT "      "
LOCATE 18, 52: PRINT "      "
LOCATE 14, 50: PRINT "         "

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "                   ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 18, 0: PRINT "MEASUREMENT IN PROGRESS, WAIT FOR COMPLETION"

TIME% = instdata(16)          ' DATA ACQUISITION PERIOD

GOSUB NORLAND

' COME BACK FROM NORLAND
' MOVE DATA INTO DATAARRAY

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "                   ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 17, 0: PRINT "DATA TRANSFER IN PROGRESS, WAIT FOR COMPLETION"
```

```
FOR x% = 1 TO 1024
        dataarray&(x%) = chanread&(x%, &HF300, &HD000)
NEXT x%


' NORMALIZE BACKGROUND COUNTS PER CHANNEL TO DATA PERIOD
' SUBTRACT NORMALIZED BACKGROUND FROM DATA
' PUT PROCESSED DATA IN PROCESSEDARRAY
' PROCESSED ARRAY IS DEMENSIONLESS IN TIME

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "           ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 18, 0: PRINT "CALCULATION IN PROGRESS, WAIT FOR COMPLETION"

FOR x% = 1 TO 1024


        processedarray!(x%) = (dataarray&(x%) - (backgroundarray&(x%) * (instdat
        IF processedarray!(x%) < 0 THEN processedarray!(x%) = 0

NEXT x%


' PERFORM AVERAGE CHANNEL CALCULATION

weightedcounts& = 0

COUNTSTOT& = 0                    ' ZERO OUT THE TOTAL

FOR I = instdata(10) TO instdata(12) ' ROI LOW TO ROI HIGH

        COUNTSTOT& = COUNTSTOT& + processedarray!(I)

        weightedcounts& = weightedcounts& + (processedarray!(I) * I)' CHANNEL FO

NEXT I

AVECHAN! = weightedcounts& / COUNTSTOT&
Zeff = (-B + SQR(B * B - 4 * A * (C - AVECHAN!))) / (2 * A)
LOCATE 14, 23: PRINT "    TOTAL COUNTS IN ROI = "; COUNTSTOT&
LOCATE 16, 23: PRINT "    AVERAGE CHANNEL NUMBER = "; USING "####.# "; AVECHAN!
LOCATE 18, 23: PRINT "    EFFECTIVE ATOMIC NUMBER = "; USING "##.##"; Zeff
'LOCATE 19, 23: PRINT ; A; B; C
BEEP
BEEP
' LOCATE 20, 23: PRINT "time/tyme ="; instdata(16) / instdata(14)
' LOCATE 13, 23: PRINT "Lowest channel# ="; instdata(10)
' LOCATE 15, 23: PRINT "Highest channel# ="; instdata(12)


FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "           ' CLEARS COMMAND LINE
```

```
NEXT I

GOTO MEASURESCREEN                          ' GET READY FOR NEXT MEASUREMENT
RETURN ' ***********************************************************
FLUSH: ' ***********************************************************
        ' *
        ' *** FLUSHES OUT THE ON SCREEN MENUS AND UPDATES DATE AND TIME ******
        ' *
        ' ***********************************************************
LOCATE 5, 3: PRINT ""; DATE$; ""
LOCATE 5, 71, 0: PRINT ""; TIME$; ""
FOR I = 7 TO 20                                    ' UPDATE REAL TIME
        LOCATE I, 3, 0: PRINT "                   ' CLEARS MENU AREA OF SCREEN
NEXT I
FOR J = 2 TO 78
        LOCATE 22, J, 0: PRINT "  "
NEXT J                                             ' CLEARS COMMAND LINE
RETURN ' ***********************************************************
CALIB: ' ***********************************************************
        ' *
        '********CALIBRATION ROUTINE ****************************
        ' ***
        ' *
        ' ***********************************************************
GOSUB FLUSH
IF BKGND = 0 THEN
   LOCATE 11, 17: PRINT " MUST HAVE BACKGROUND SPECTRUM BEFORE TAKING DATA."
   LOCATE 22, 18, 0: PRINT " HIT ANY KEY TO RETURN TO MAIN MENU."
   DO
        LOCATE 5, 71, 0: PRINT ""; TIME$; ""
   LOOP WHILE INKEY$ = ""
   GOTO MENU
END IF

' *** MEASUREMENT SCREEN DISPLAY ***

LOCATE 8, 26: PRINT "*** CALIBRATION MODE ***"
LOCATE 12, 11: PRINT " [ ]  USE DEFAULT CALIBRATION"
LOCATE 14, 11: PRINT " [ ]  ENTER NEW CALIBRATION CONSTANTS"
LOCATE 16, 11: PRINT " [ ]  RECALIBRATE"
LOCATE 22, 14: PRINT " USE <UP> OR <DOWN> ARROWS THEN <ENTER TO SELECT. "
       D1 = 12

LIM1: ' *** MANAGES THE LIMITS OF TRAVEL OF THE CURSOR **********************
       IF D1 > 16 THEN D1 = 12 ELSE IF D1 < 12 THEN D1 = 16
       LOCATE D1, 13, 1, 0, 13
       D1 = CSRLIN
       A$ = INKEY$: IF A$ = CHR$(13) THEN GOTO MCURSVAL1 ELSE IF A$ = "8" THEN
       IF A$ = "2" THEN D1 = D1 + 2: GOTO MLIM1
```

```
        IF LEN(A$) = 2 THEN A$ = RIGHT$(A$, 1) ELSE GOTO MLIM1
        IF A$ = "H" THEN D1 = D1 - 2: GOTO MLIM1
        IF A$ = "P" THEN D1 = D1 + 2: GOTO MLIM1
        GOTO MLIM1
        D1 = D1 + 2: GOTO MLIM1

MCURSVAL1: ' *** FINDS THE VALUE OF THE LINE AT THE CURSOR LOCATION ************
        D1 = CSRLIN
        IF CSRLIN = 12 THEN GOSUB DEFAULT               ' USE DEFAULT CALIBRATION
        IF CSRLIN = 14 THEN GOSUB NEWCON                ' ENTER NEW CONSTANTS A, B, C.
        IF CSRLIN = 16 THEN GOSUB RECAL                 ' RECALIBRATE
        GOTO MENU
RETURN
DEFAULT:
GOSUB FLUSH
LOCATE 10, 19: PRINT "DEFAULT CALIBRATION CONSTANT VALUES"
LOCATE 13, 31: PRINT "A = -0.206"
LOCATE 15, 31: PRINT "B = 14.072"
LOCATE 17, 31: PRINT "C = 27.189"
LOCATE 22, 18: PRINT "HIT <ESC> TO RETURN TO MAIN MENU."
CAL = 1
DO
  A$ = INKEY$
  IF A$ = CHR$(27) THEN GOTO MENU
  LOCATE 5, 71, 0: PRINT ""; TIME$; ""
LOOP UNTIL A$ = CHR$(27)
RETURN
NEWCON:
GOSUB FLUSH
LOCATE 12, 11: INPUT " A = ? ", A
LOCATE 14, 11: INPUT " B = ? ", B
LOCATE 16, 11: INPUT " C = ? ", C
CAL = 1
LOCATE 22, 18: PRINT "HIT <ESC> TO RETURN TO MAIN MENU."
DO
  A$ = INKEY$
  IF A$ = CHR$(27) THEN GOTO MENU
  LOCATE 5, 71, 0: PRINT ""; TIME$; ""
LOOP UNTIL A$ = CHR$(27)
RETURN

RECAL:
GOSUB FLUSH
LOCATE 10, 22: PRINT "MEASUREMENT PERIOD IS "; instdata(18); "SECONDS"
LOCATE 22, 3, 0: PRINT "HIT <ESC> TO USE DEFAULT CALIBRATION. HIT ANY OTHER KEY
DO
  A$ = INKEY$
  IF A$ = CHR$(27) THEN GOTO MENU
  LOCATE 5, 71, 0: PRINT ""; TIME$; ""
LOOP WHILE A$ = ""
FOR J = 2 TO 78
  LOCATE 22, J, 0: PRINT "  "
NEXT J
LOCATE 13, 3, 0: INPUT "        NUMBER OF CALIBRATION POINTS = ? (MUST BE >2) ", N
FOR M = 1 TO N
```

```
' LOCATE 20, 8, 0: PRINT " THE REMAINING NUMBER OF CALIBRATION POINTS IS "; N
  LOCATE 17, 3, 0: INPUT "      EFFECTIVE Z OF TARGET IS ? ", Z(M)
LOCATE 22, 3, 0: PRINT "POSITION PROBE, THEN HIT <SPACE BAR> TO BEGIN MEASUREMEN

DO
        LOCATE 5, 71, 0: PRINT ""; TIME$; ""          ' UPDATE REAL TIME

A$ = INKEY$
        IF A$ = CHR$(27) THEN GOTO MENU          ' INTERRUPT
        IF A$ = CHR$(32) THEN GOTO CALOK
LOOP

CALOK:

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "              ' CLEARS COMMAND LINE
        LOCATE 17, I, 0: PRINT " "
NEXT I

LOCATE 22, 18, 0: PRINT "MEASUREMENT IN PROGRESS, WAIT FOR COMPLETION"

TIME% = instdata(18)        ' DATA ACQUISITION PERIOD

GOSUB NORLAND

' COME BACK FROM NORLAND
' MOVE DATA INTO DATAARRAY

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "              ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 17, 0: PRINT "DATA TRANSFER IN PROGRESS, WAIT FOR COMPLETION"

FOR x% = 1 TO 1024
        dataarray&(x%) = chanread&(x%, &HF300, &HD000)
NEXT x%


' NORMALIZE BACKGROUND COUNTS PER CHANNEL TO DATA PERIOD
' SUBTRACT NORMALIZED BACKGROUND FROM DATA
' PUT PROCESSED DATA IN PROCESSEDARRAY
' PROCESSED ARRAY IS DIMENSIONLESS IN TIME

FOR I = 2 TO 78
        LOCATE 22, I, 0: PRINT "  "              ' CLEARS COMMAND LINE
NEXT I

LOCATE 22, 18, 0: PRINT "CALCULATION IN PROGRESS, WAIT FOR COMPLETION"

FOR x% = 1 TO 1024


        processedarray!(x%) = (dataarray&(x%) - (backgroundarray&(x%) * (instdat
        IF processedarray!(x%) < 0 THEN processedarray!(x%) = 0
```

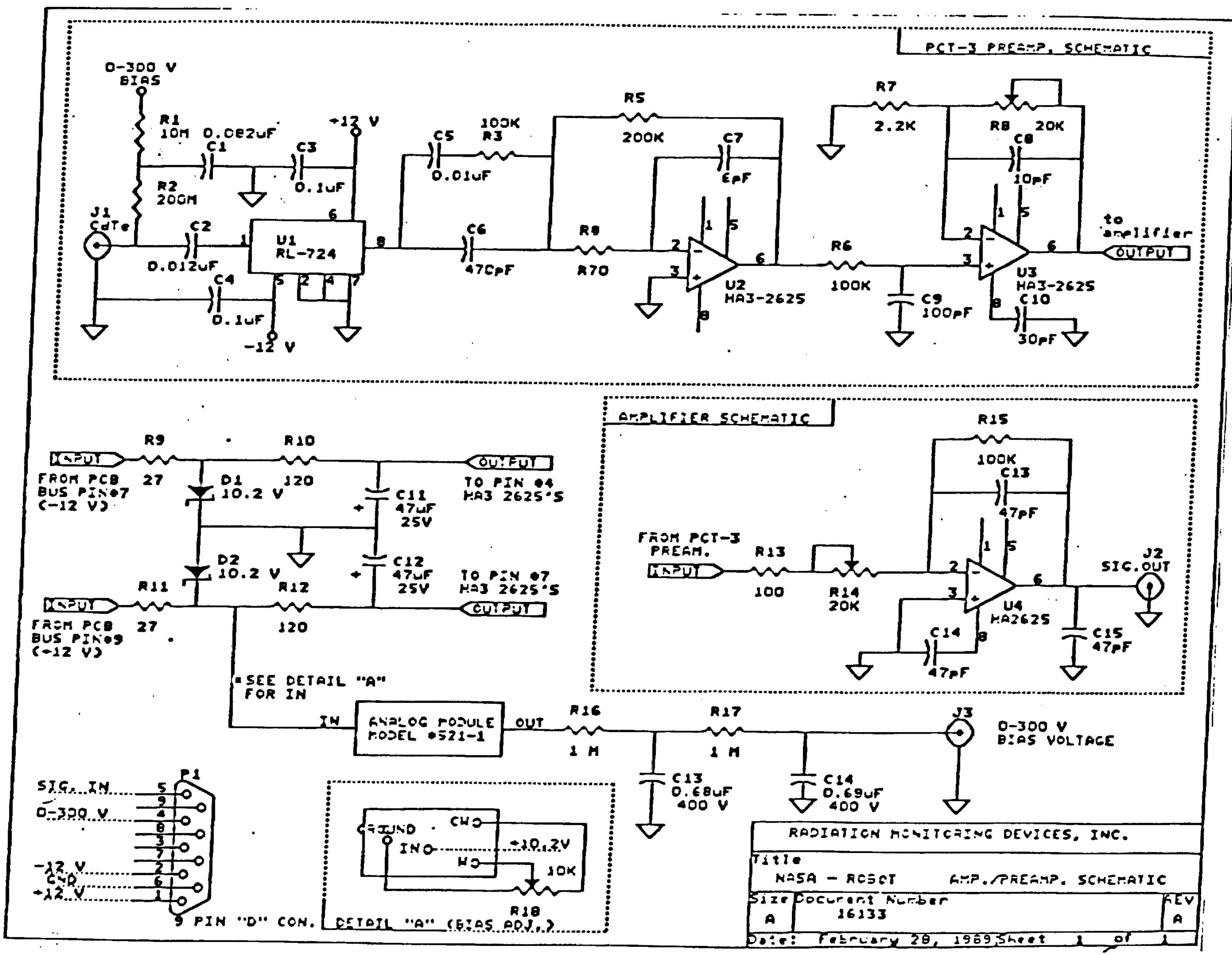
```
NEXT x%

' PERFORM AVERAGE CHANNEL CALCULATION

weightedcounts& = 0

COUNTSTOT& = 0                    ' ZERO OUT THE TOTAL

FOR I = instdata(10) TO instdata(12) ' ROI LOW TO ROI HIGH

        COUNTSTOT& = COUNTSTOT& + processedarray!(I)

        weightedcounts& = weightedcounts& + (processedarray!(I) * I)' CHANNEL FO

NEXT I

AVECHAN! = weightedcounts& / COUNTSTOT&
E(M) = AVECHAN!
BEEP
FOR I = 2 TO 78
    LOCATE 22, I, 0: PRINT " "
    LOCATE 17, 3, 0: PRINT " "
NEXT I

NEXT M
' Least-Squares Calculation Routine .
        'A  program to calculate the coefficients for a least-squares
        'fit of a second-order polynomial:
        '   <ch#> = A*Z^2 + B*Z + C
        'to calibration data points.

        'Least-Squares Calculation

        S1 = 0: S2 = 0: S3 = 0: S4 = 0: S5 = 0: S6 = 0: S7 = 0

        FOR I = 1 TO N
            S1 = S1 + Z(I)
            S2 = S2 + Z(I) * Z(I)
            S3 = S3 + Z(I) ^ 3
            S4 = S4 + Z(I) ^ 4
            S5 = S5 + E(I)
            S6 = S6 + Z(I) * E(I)
            S7 = S7 + Z(I) * Z(I) * E(I)
        NEXT I

        D1 = N * (S2 * S4 - S3 * S3)
        D2 = S1 * (S2 * S3 - S1 * S4)
        D3 = S2 * (S1 * S3 - S2 * S2)
        D = D1 + D2 + D3

        A1 = N * (S2 * S7 - S3 * S6)
        A2 = S1 * (S2 * S6 - S1 * S7)
        A3 = S5 * (S1 * S3 - S2 * S2)
```

```
B1 = N * (S6 * S4 - S3 * S7)
B2 = S5 * (S2 * S3 - S1 * S4)
B3 = S2 * (S1 * S7 - S2 * S6)

B = (B1 + B2 + B3) / D
A = (A1 + A2 + A3) / D


C1 = S5 * (S2 * S4 - S3 * S3)
C2 = S1 * (S3 * S7 - S4 * S6)
C3 = S2 * (S6 * S3 - S7 * S2)

C = (C1 + C2 + C3) / D

FOR I = 2 TO 78
    LOCATE 22, I, 0: PRINT "  "
    LOCATE 20, I, 0: PRINT "  "
NEXT I
LOCATE 10.22: PRINT "
LOCATE 13.3: PRINT "

LOCATE 12, 12: PRINT "CALIBRATION COEFFICIENTS"
LOCATE 14, 12: PRINT "A = "; A
LOCATE 15, 12: PRINT "B = "; B
LOCATE 16, 12: PRINT "C = "; C
LOCATE 12, 40: PRINT "Z(TRUE)"
LOCATE 12, 50: PRINT "Z(CALCULATED)"
FOR I = 1 TO N
    LOCATE 13 + I, 40: PRINT Z(I)
    LOCATE 13 + I, 50: PRINT (-B + SQR(B * B - 4 * A * (C - E(I)))) / (2
NEXT I

CAL = 1
LOCATE 22, 3, 0: PRINT "CALIBRATION IS COMPLETE. PRESS <ESC> TO RETURN T
DO
A$ = INKEY$
IF A$ = CHR$(27) THEN GOTO MENU
LOOP


RETURN

NORLAND: ' ***********************************************************
         ' *
         ' *** NORLAND MCA-5000 CONTROL ROUTINES ***************************
         ' *
         ' *****************************************************************
' PHA Initialization:

' This is for I/O address jumpers set for F300h, IRQ7, D0000h
' This example is for PHA mode, ADD, DIR, 1k gain, lld = 20, uld = 1024,
' offset = 0, 1 sec timebase (LIVEtime), 1k memory size.
gain% = 1024
```

```
CALL setoffset(0, gain%)                    ' set 0 offset and 1k gain
CALL setlld(20)                             ' SET LLD TO CHANNEL 20
CALL setuld(1023)                           ' SET ULD TO CHANNEL 1023

OUT &HF307, &HD5                            ' now write mode register

        ' amp mode - bit 7 =0, PHA mode - bit 6 set,
        ' lower memory bank - bit 5 =0, ADD - bit 4 set
        ' 1st 1k memory, bits 3-0 are 0101
        ' FOR AMP MODE USE &H55 INSTEAD OF &HD5

OUT &HF304, &H8C              ' 100ms base rate * 10

CALL erasepha(4096, &HF0, &HF300, &HD000)   ' clear the memory before starting ac
CALL acquire(TIME%, &HF2, &HF300)           ' acquire TIME% sec realtime

' FOR REAL TIME USE CALL acquire(TIME%, &HF0, &HF300)

RETURN ' ************************************************************************

DEFLNG A-Z
FUNCTION chanread (A%, B%, C%)
OUT B%, &H41
DEF SEG = C%
        temp0 = PEEK(A% * 4)
        temp1 = PEEK(A% * 4 + 1)
        temp2 = PEEK(A% * 4 + 2)
chanread = temp2 * 65536 + temp1 * 256 + temp0

END FUNCTION

SUB setlld (lld%)
SHARED gain%

' This routine will set up the lower level discriminator
' Registers 0 and 7 are affected and must be rewritten after this routine

' value written to registers depends on conversion gain

lld% = lld% * 4096! / gain%

upperbyte% = FIX(lld% / 256)              ' separate upper byte
lowbyte% = FIX(lld% MOD &HFF)             ' separate lower byte
OUT &HF307, upperbyte%                    ' put upper 4 bits of lld in Register7
OUT &HF300, &H80                          ' enable lld register and discriminator buffe
OUT &HF301, lowbyte%                      ' write to lld register.  The upper 4
                                                      ' bits from reg7 are als
OUT &HF300, &H40                          ' now disable discriminator registers
END SUB

SUB setoffset (offset%, gain%)
```

```
' This routine will set the proper offset and will adjust
' factory offset for different gains
' The conversion gain is also set by this routine

DEFLNG V

' First get factory offset

IF gain% = 8192 THEN

' A note about the hexadecimal numbers:
' if I just specify &h0000ff42 for a long integer, QuickBasic will
' chop off the leading 0's when i move to the next line.  When it
' interprets the number, it is sign extended, leaving ffffff42, which
' I don't want. So I have 2 alternatives: 1. Convert the number to
' decimal, 65346, and specify in decimal or 2. Do something like
' this: &H1000ff42 AND &H2000ff42 which will be interpreted as 0000ff42
' The second way is more readable from a programming point of view, but
' the first method will be faster in execution speed.

    VALUE& = &H1000FEE9 AND &H2000FFFF
        ' factory offfset 3fe9 + gain bits 11
ELSEIF gain% = 4096 THEN
        VALUE& = &H1000BF65 AND &H2000FFFF
        ' factory offset 3f65 + gain bits 10
ELSEIF gain% = 2048 THEN
        VALUE& = &H7FA4
        ' factory offset 3fa4 + gain bits 01
        ' since there is no sign bit, the 7fba is read correctly
ELSEIF gain% = 1024 THEN
        VALUE& = &H3FC3
        ' factory offset 3fc3 + gain bits 00
END IF

' Now subtract desired offset
VALUE& = VALUE& - offset%

' now write the registers

lowvalue% = VALUE& AND &HFF              ' get lower byte
hivalue% = FIX(VALUE& / 256)             ' separate upper byte

OUT &HF302, lowvalue%                    ' output offset (0 + factory offset for 8k)
OUT &HF303, hivalue%                     ' upper offset 3FH + gain bits 2K
OUT &HF305, 0                            ' load the offset into the hardware

END SUB

SUB setuld (uld%)
SHARED gain%

' This routine will set up the upper level discriminator
' Registers 0 and 7 are affected and must be rewritten after this routine
' value written to registers depends on conversion gain
'
```

```
uld% = uld% * 4096! / gain%

upperbyte% = FIX(uld% / 256)          ' separate upper byte
lowbyte% = FIX(uld% AND &HFF)         ' separate lower byte
OUT &HF307, upperbyte%                ' put upper 4 bits of uld in Register7
OUT &HF300, &H0                       ' enable uld register and discriminator buffe
OUT &HF301, lowbyte%                  ' write to uld register.  The upper 4
                                                              ' bits from reg7 are als
OUT &HF300, &H40                      ' now disable discriminator registers

END SUB
```



PCT-3 PREAMP. SCHEMATIC
0-300 V BIAS
R1 10M
C1 0.082uF
C3 0.1uF
+12 V
R2 200M
J1 CdTe
C2 0.012uF
U1 RL-724
C4 0.1uF
-12 V
C5 0.01uF
100K R3
R5 200K
C7 6pF
C6 47CpF
R8 R70
U2 HA3-2625
R6 100K
C9 100pF
R7 2.2K
R8 20K
C8 10pF
U3 HA3-2625
C10 30pF
to amplifier
OUTPUT
AMPLIFIER SCHEMATIC
R15 100K
C13 47pF
FROM PCT-3 PREAM.
INPUT
R13 100
R14 20K
U4 HA2625
C14 47pF
C15 47pF
J2 SIC.OUT
R9 27
R10 120
INPUT
FROM PCB BUS PIN#7 (-12 V)
D1 10.2 V
C11 47uF 25V
OUTPUT
TO PIN #4 HA3 2625'S
D2 10.2 V
C12 47uF 25V
TO PIN #7 HA3 2625'S
R11 27
R12 120
FROM PCB BUS PIN#9 (-12 V)
*SEE DETAIL "A" FOR IN
IN
ANALOG MODULE MODEL #521-1
OUT
R16 1 M
R17 1 M
C13 0.68uF 400 V
C14 0.69uF 400 V
J3
0-300 V BIAS VOLTAGE
P1
SIG. IN
0-300 V
-12 V
GND
+12 V
9 PIN "D" CON.
GROUND
IN
CH
W
+10.2V
10K
R18
DETAIL "A" (BIAS ADJ.)
RADIATION MONITORING DEVICES, INC.
Title
NASA - RCSCT AMP./PREAMP. SCHEMATIC
Size A
Document Number 16133
REV A
Date: February 28, 1989 Sheet 1 of 1

APPENDIX A

A SYSTEM FOR PRECISE DETERMINATION OF EFFECTIVE ATOMIC NUMBER BY BETA BACKSCATTER

Authors:
M. S. Soller, L. Cirignano, P. Lieberman, and M.R. Squillante

IEEE Transactions on Nuclear Science
Vol. 37, No. 2, 230-233, April 1990

IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 37, NO. 2, APRIL 1990

A SYSTEM FOR PRECISE DETERMINATION OF EFFECTIVE ATOMIC NUMBER BY BETA BACKSCATTER

M.S. Soller, L. Cirignano, P. Lieberman, and M.R. Squillante
Radiation Monitoring Devices, Inc.
44 Hunt Street, Watertown, MA 02172
(617) 926-1167

Abstract

A compact instrument capable of determining the effective atomic number (Z_{eff}) of a target has been developed and characterized. It is a non-destructive sensor which utilizes a cadmium telluride (CdTe) solid state detector and a $^{90}Sr/^{90}Y$ source in a backscatter geometry. The spectrum of the backscattered beta particles depends on the target atomic number. Measurements show that by using the mean energy of the backscatter spectrum, the average effective atomic number can be determined to better than ± 1 Z.

INTRODUCTION

In many applications which use ionizing radiation as a probe, the size, weight and power requirements of the nuclear sensor are important considerations, particularly for portable instruments and instruments designed for use in space. With this in mind we undertook the development of a radiation-backscatter sensor system which is small, lightweight and uses little power. The design of the system can be adapted for a variety of non-destructive analysis applications by selecting the type and energy of the radiation source used. The system described here was designed to determine the average Z of a target by measuring beta backscatter.

Beta backscatter is largely caused by coulomb scattering from the target nucleus [1,2]. As a result, the spectrum of backscattered betas depends on the target atomic number. This was demonstrated in an early experiment by Brownell [3] who showed that the average energy of the backscattered spectrum increases monotomically with target Z. We have exploited this relationship to determine the Z_{eff} of target materials.

The system developed includes a sensor head (Figures 1 and 2), electronics and software. The sensor head contains both the radiation detector and a radioactive isotope. The sensor chosen is a CdTe particle detector we manufacture. The source is a 2 mCi $^{90}Sr/^{90}Y$ ceramic pellet. This ceramic source is rugged and safe, preventing potential contamination. The maximum energy of the betas is 2.3 MeV.

The source is embedded into a threaded plastic piece which screws into the sensor head, allowing for adjustment of the beam collimation. The beam collimation is adjustable from a beam spread of 15° to 50°. The field of view of the detector is larger than the area illuminated by the radiation at any collimation, which allows the beam collimation to define the target area of analysis.

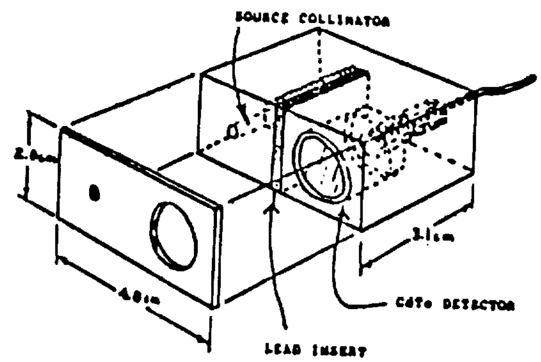


Figure 1. Diagram of sensor head for beta backscatter measurements.

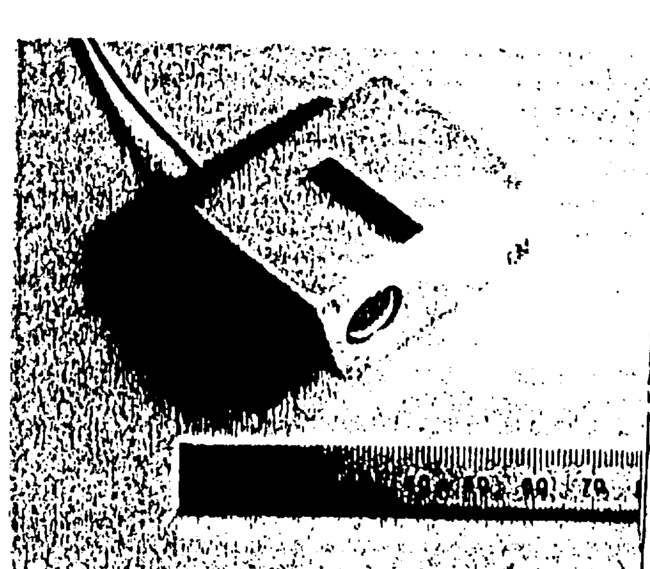


Figure 2. Photograph of sensor head for beta backscatter measurements.

Because of the close proximity of the source and detector, background noise from the source is a concern. Several steps were taken to minimize the background while maintaining the desired small size. When using beta emitting sources, background noise is due to both the direct beta particles themselves and to the bremstrahlung radiation generated as the particles pass through matter.

The elimination of the noise due to the direct detection of the beta particles simply required that the minimum wall thicknesses be equal to or greater than the maximum range of the betas. The wall thicknesses will vary depending on the material used. This, however, is not the most important source of noise.

A more difficult problem stems from the fact that beta particles generate bremsstrahlung radiation as they pass through matter. The magnitude of the bremsstrahlung flux produced is a linear function of Z; minimizing Z minimizes the flux of bremstrahlung radiation. Thus, the sensor head is made from a low Z plastic. There is still a significant amount of background from the bremstrahlung generated by the plastic. Reducing this to acceptable levels required that about 4 mm of lead be added between the sensor and source.

A final consideration in the choice of material used to make the sensor head is that many plastics degrade under constant exposure to radiation. To address this, Delrin™ plastic was chosen because it is relatively insensitive to radiation.

Electronics

The electronic circuitry for the system consists of an RMD, Inc., PCT3 preamplifier and an RMD, Inc., KS2 PC-board linear amplifier with bias supply which is inserted into the PC. The output of the linear amplifier is sent to a Norland 5000™ PC-board MCA. The linear amp, detector bias supply, and MCA are all installed in a portable personal computer. The detector signal is coupled to the charge sensitive preamplifier via six feet of small diameter, low noise shielded cable.

The computer controls the calibration, data acquisition, and data analysis. A series of menus leads the user through calibration, collection of a reference background and collection of data. The MCA stores the data in RAM. Background is subtracted from the spectrum, then the data is put through a simple weighted mean algorithm. This produces a mean channel number (each channel represents an energy bin). The calibration routine does this for several known samples and calculates the constants for a third order polynomial fit.

To analyze a target material, the sample is scanned and the mean channel number from the unknown target compared to the calibration curve. Based on the result a Z_{eff} value is assigned to it. This is displayed to the user along with an uncertainty value based on the total integrated counts from the measured spectrum. In the final design, a look-up table will be added, to identify likely materials which fit these parameters.

Experimental Set-up

In operation, the face of the sensor head is positioned so that the collimated beam irradiates the target area of interest. The close proximity of the detector to the source gives a backscatter angle of approximately 180°, as long as the target distance is greater than the minimum defined by the system. The cone of the collimated beam and the sensor head to target distance determine the area of interrogation and the flux of backscattered betas viewed by the detector.

In order to get reproducible results, there are several factors which need to be considered before making measurements. The radiation beam takes the form of a cone, which irradiates an increasing area with increasing sensor-to-target distance. The target must be large enough to fill the entire field of interrogation at the distance and collimation being used. The targets may have uneven or irregular shapes, but must be of uniform composition. Finally, the targets must effectively be "infinitely thick" to the radiation. According to measurements made by Sharma and Singh [4] and confirmed by measurements made in this study, infinitely thick is approximately equal to 20% of the maximum range of the maximum energy of the beta emissions being used. For the $^{90}Sr/^{90}Y$ source being used (maximum energy = 2.3 MeV), this translates to less than 1 mm for targets with densities greater than 1.5 g/cm^3. All targets used in this study were infinitely thick.

Another factor which must be taken into account is the effect of air on the beta particles. Although the range of the 2.3 MeV betas is more than 7 meters in air, a significant fraction of the lower energy betas can experience coulomb scattering and energy loss through ionization in distances as short as a few centimeters. This changes the backscattered spectrum as a function of distance. Additionally, scattering of the lower energy betas in air causes an expansion of the beam cone. Targets which otherwise would be large enough to fill the field of the beam may be too small as a result of the scattering of low energy betas in air. When a target is too small, the low energy betas which have air scattered to the outer edges of the beam miss the target and are not backscattered. This loss of low energy betas from the backscattered signal shifts the mean energy upwards.

Experimental Results

The CdTe detector was operated with the negative contact irradiated. Since the beta particles do not completely penetrate the CdTe, and since the electron mobility-lifetime product is higher than that for holes, this orientation provides better sensitivity to the betas. With a 2 meter cable, the noise threshold of the system was about 25 keV.

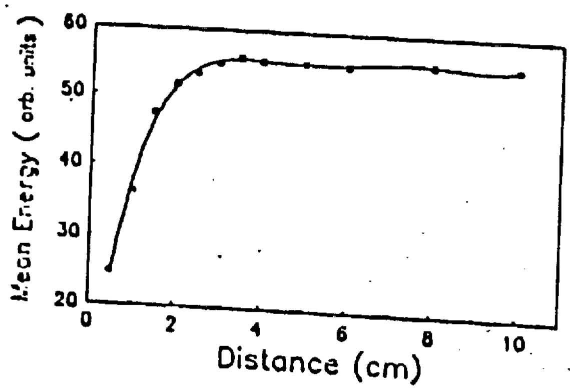


Figure 3. Determination of minimum operating distance of sensor head from target.

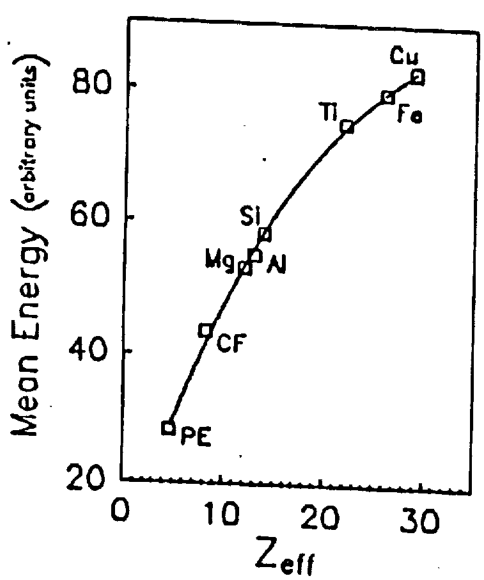


Figure 4. The sensitivity of the beta backscatter device to the Z_{eff} is shown to be better than ±1 Z.

The sensor system performance was characterized by measuring the following parameters:

1) mean energy as a function of sensor-to-target distance
2) mean energy as a function of Z_{eff}
3) mean energy as a function of density
4) the effect of varying detector bias voltage.

The minimum operating distance was determined empirically. Using an aluminum target, spectra were collected at distances ranging from 0.5 cm to 10 cm. The mean energy was determined and plotted as a function of distance. As can be seen in Figure 3, when the distance is less than about 3.5 cm, angular effects cause the backscatter angle to deviate sufficiently from 180° to alter the backscatter spectrum. Therefore, 3.5 cm was used as the minimum operating distance of the device.

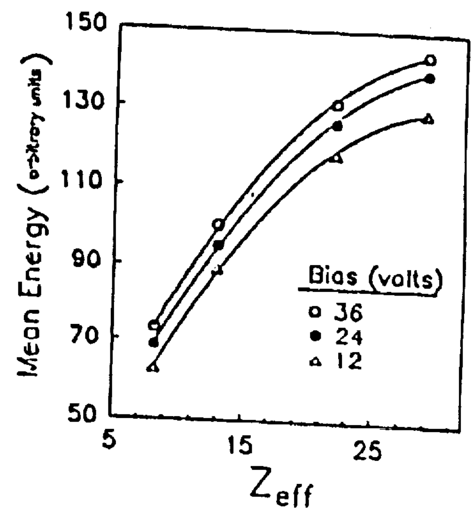


Figure 5. The effect of varying the detector bias on sensitivity to Z_{eff}. 24 volts was chosen as the operating bias.

The effective atomic number of a target material determines the mean energy of backscattered beta particles. Measurements were made to determine the sensitivity of the sensor head to changes in Z_{eff}. Several pure elements as well as binary compounds were tested. Figure 4 shows the uncertainty of Z_{eff} to be about ±1 Z. Repeated measurements indicate that the uncertainty in the calculated Z_{eff} of the target is slightly larger than $N^{-1/2}$ where N is the total integrated counts. This holds for all N until the Z_{eff} uncertainty reaches near 1. Increasing the counts does not improve the accuracy significantly. Extrapolation of the data gives an ultimate accuracy of ±0.5 Z.

Measurements were made to determine the effect of bias on the mean energy calculation being used to determine Z_{eff}. CdTe detectors can operate over a wide range of bias voltages. For the 1 mm device chosen for these measurements, the bias could be varied from about 5 volts to more than 60 volts. Higher bias improves sensitivity to higher energy radiations but increases the noise. The results (see Figure 5) indicate that above a threshold of about 20 volts, there is no net effect of bias on the determination of Z_{eff}. Therefore, 24 volts was chosen as the operating bias since this is above the threshold and this voltage is readily available in most applications.

The influence of density on the backscatter signal was also examined. The oxides of beryllium and hydrogen shed useful light on this question, since they have identical Z_{eff}, but the beryllium compound is 2.9 times more dense than the water. The mean energy of the backscattered betas is virtually identical for both targets. Moreover, the trio of graphite, silicon, and potassium chloride are of very similar densities; 2.2, 2.33, and 1.98, respectively. However, the mean channel numbers of their spectra, 34.2, 59.4, and 76.2, mirror the progression in their Z_{eff}: 6, 14, 18.3.

These observations suggest the conclusion that density does not influence the mean energy of the backscattered betas. Additionally, the smoothness of the curve in Figure 4 supports this conclusion; since the densities of materials with increasing Z_{eff} fluctuate dramatically.

Conclusions

A compact backscatter analysis system was developed based on a CdTe detector and an isotopic beta source. The system includes data acquisition and analysis software which runs on a personal computer. Testing indicates that it was possible to distinguish the average atomic number of a material to ±1 Z regardless of the density of the material. Further improvements promise to give better than ±0.5 Z sensitivity. The system was designed to be flexible enough to be applied to a variety of applications by changing the data analysis algorithms and the source type and energy.

References

[1] Gray, P.R., Clarey, D.H. and Beamaer, W.H., *Interaction of Beta Particles with Matter: Analysis of Hydrocarbons by Beta-ray Backscattering*, 31, Dec. (1959).

[2] Knoll, G.F., *Radiation Detection and Measurement*, 2nd Ed., Wiley, NY., p 348.

[3] Brownell, G.L., *Interactions of P-32 Beta Rays with Matter*, Nucleonics, 10 No. 8, 30 (1952).

[4] Sharma, K.K. and Singh, M., J. Appl. Phys., 50, No. 3 1529 (1979).

NASA

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
09		

4. Title and Subtitle	5. Report Date
Novel, Proportional Proximity Sensor for Autonomous Space Based Robots - Phase II	July 1990
	6. Performing Organization Code: C88-8
7. Author(s): M.R. Squillante, Ph.D., L.P. de Rochemont, M. Eng., L. Cirignaho, M.S., P. Lieberman, Ph.D. and M.S. Soller, M.S.	8. Performing Organization Report No.: 09
	10. Work Unit No.
9. Performing Organization Name and Address: Radiation Monitoring Devices, Inc., 44 Hunt Street, Watertown, MA 02172	11. Contract or Grant No.: NAS7-1032
	13. Type of Report and Period Covered: Final Report 3/2/88 - 7/2/90
12. Sponsoring Agency Name and Address: NASA RESIDENT OFFICE - JPL, 4800 Oak Grove Drive, Pasadena, CA 91109	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The goal of this program was to develop new sensing capabilities for autonomous robots operating in space. Information gained by the robot using these new capabilities would be combined with other information gained through more traditional capabilities, such as video, to help the robot characterize its environment as well as to identify known or unknown objects that it encounters. Several sensing capabilities using nuclear radiation detectors and backscatter technology were investigated.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
robotic sensor, composition sensor, radiation backscatter	Mr. P. Tackney, Contract Officer Mr. J. Goudy, NASA Tech Monitor Dr. P. McGuire, JPL Mgr. SBIR Prog. Dr. N. Marzwell, JPL Tech Monitor NASA Sci. & Tech. Info.Facility

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	36	